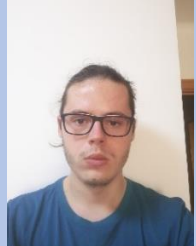
	M5 - ENTORNOS DE DESARROLLO		
	UF2. Optimización de software		
UF2-Pr4 Refactorización de código			
Apellidos: Farré Ruiz	Nombre: Pau	Fecha	13/01/2023

Contenido

Refactorización del Código del Videojuego.....	2
1. Usar las variables/parámetros por referencia para crear funciones globales donde creáis oportuno de vuestro código. (5p).....	2
2. Modificar el código para mejorarlo, usando lo aprendido en M3 y M5/M16 desde el desarrollo de la PR3 (3p)	2
3. Subir a GIT el juego, teniendo en cuenta los parámetros del apartado de Entrega de la Actividad PR4. (2p)	2

Refactorización del Código del Videojuego

1. Usar las variables/parámetros por referencia para crear funciones globales donde creáis oportuno de vuestro código. (5p)

En mi código tengo dos parámetros por referencia, el primero:

```
/*parametro por referencia*/  
void enemyAttack(int& enemyDamage) {  
  
    enemyDamage = 250 + rand() % (1500 - 0);  
  
    if (enemyIsAlive1 == false)  
    {  
        enemyDamage1 = 0;  
    }  
    else if (enemyIsAlive2 == false)  
    {  
        enemyDamage2 = 0;  
    }  
}
```

En este caso lo que hice fue tener una sola función que me da el daño del enemigo y no tener una por cada enemigo.

Y el segundo:

```
/*parametro por referencia*/  
void enemyHP(int& enemyHP) {  
  
    enemyHP = enemyHP - heroDamage;  
}
```

En este caso lo que hice fue tener una sola función para que me diga la vida que le queda al enemigo y no tener una por enemigo.

2. Modificar el código para mejorarlo, usando lo aprendido en M3 y M5/M16 desde el desarrollo de la PR3 (3p)

No modifique le código usando lo aprendido en M3 y M5/M16 ya que ya lo hice cuando se crearon las funciones optimizando el código lo máximo posible.

3. Subir a GIT el juego, teniendo en cuenta los parámetros del apartado de Entrega de la Actividad PR4. (2p)